

**Universidad Católica Andrés Bello**  
**Escuela de Ingeniería informática**  
**Cátedra de Algoritmos y Programación I**

**Proyecto Final Grupal (Juega y crea conciencia sobre los ODS), primera entrega.**

**Alumnos:**

**César Barreto**

**Edwin Rodríguez**

**Juan Quijada**

**Julio Campos**

**Docentes:**

**Franklin Bello**

**Zulma Díaz**

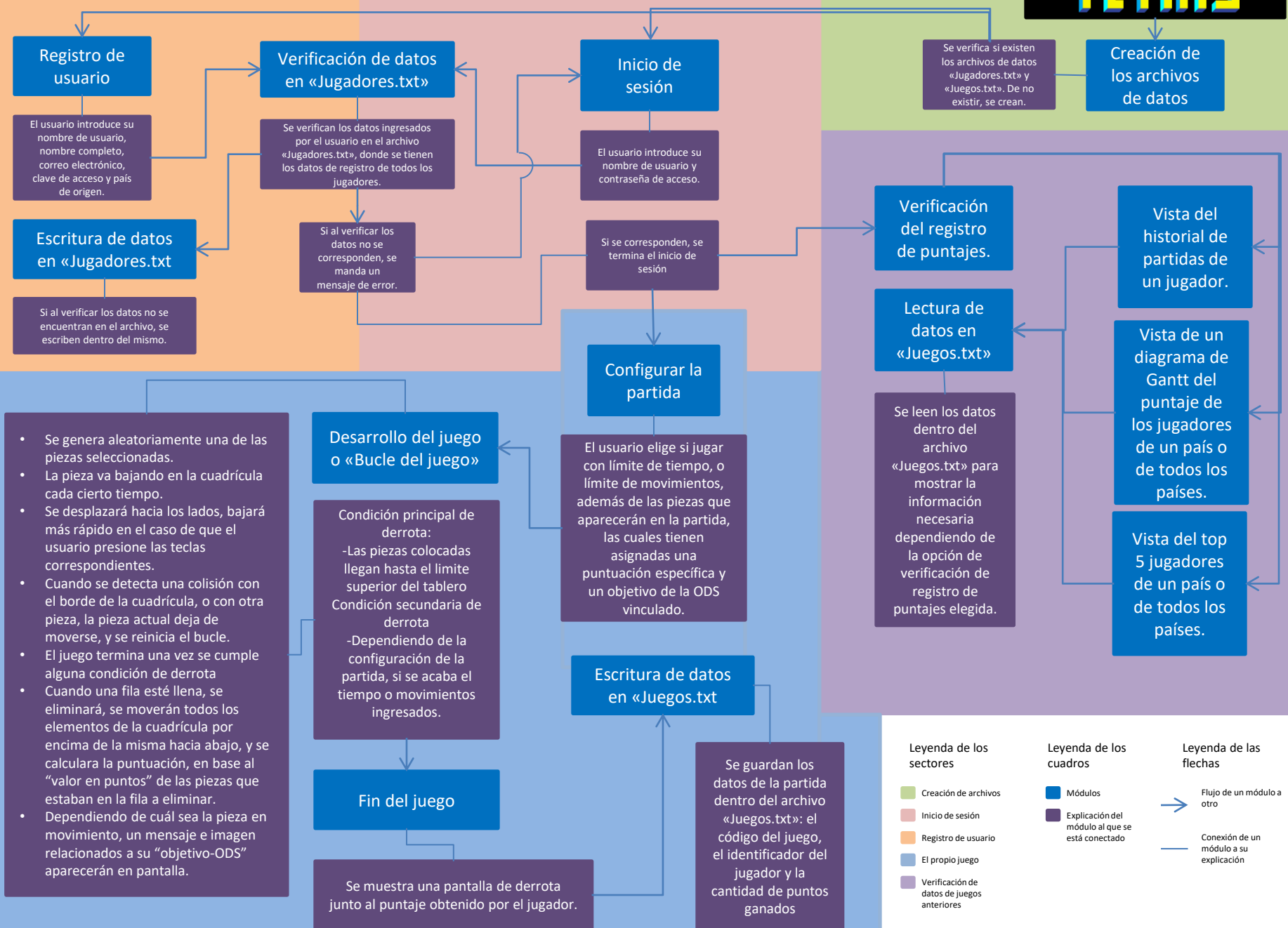
**Puerto Ordaz 26/06/2022**

## 1.- Análisis del problema (presentando entradas, procesos, salidas)

Entrada	Procesos	Salida
<ol style="list-style-type: none"> <li>Las opciones del usuario dentro del menú principal (Registrarse, iniciar sesión, cerrar sesión, Jugar o ver los datos de partidas pasadas o de otros usuarios, salir del juego.)</li> <li>Los datos del usuario a la hora de registrarse (Nombre de usuario, Nombre completo, Clave de seguridad, Correo electrónico y país de procedencia)</li> <li>Los datos del usuario a la hora de iniciar sesión (Nombre de usuario y Clave de seguridad)</li> <li>Las opciones del usuario dentro del menú de configuración del juego (Elegir el modo de juego (Contrarreloj o por movimientos) y las piezas a utilizar dentro del juego) <ol style="list-style-type: none"> <li>El tiempo máximo de juego si es contrarreloj.</li> <li>La cantidad de movimientos permitidos si es por movimientos.</li> </ol> </li> <li>La dirección de movimiento de la pieza mientras se ejecuta el propio juego (Derecha, izquierda o abajo)</li> <li>Las opciones del usuario dentro del menú de los datos de juegos pasados o de otros usuarios (Ver diagrama de Gantt de los puntos de los jugadores de un país o de todos los jugadores, ver un reporte de los datos de un jugador, ver un reporte del top 5 jugadores de todos los países o de un país en particular)</li> <li>Los datos del usuario del jugador del que se quiere el reporte (Nombre de usuario)</li> <li>El país del que se quiere ver el diagrama de Gantt o el top 5.</li> </ol>	<ol style="list-style-type: none"> <li>Se verifica si existe el archivo que contiene los datos de los usuarios (JUGADORES.TXT) y el archivo que contiene los datos de partidas pasadas. (JUEGOS.TXT) <ol style="list-style-type: none"> <li>Si no existe alguno, se crea el archivo correspondiente desde cero.</li> </ol> </li> <li>Se pide la opción al usuario del menú principal <ol style="list-style-type: none"> <li>Registrarse <ol style="list-style-type: none"> <li>Se piden los datos del usuario a la hora de registrarse</li> <li>Se realizan las verificaciones necesarias para revisar si los datos ingresados son válidos</li> <li>Si se cumplen los requisitos necesarios, se escriben los datos en el archivo JUGADORES.TXT</li> </ol> </li> <li>Iniciar sesión <ol style="list-style-type: none"> <li>Se piden los datos del usuario a la hora de iniciar sesión</li> <li>Se verifica si los datos existen y se corresponden entre sí.</li> <li>Se mantiene la sesión dentro del programa.</li> </ol> </li> <li>Cerrar sesión <ol style="list-style-type: none"> <li>Se cierra la sesión iniciada anteriormente dentro del programa.</li> </ol> </li> <li>Jugar <ol style="list-style-type: none"> <li>Se piden los datos del usuario dentro del menú de configuración del juego</li> <li>Se guarda durante la partida el tipo de juego y piezas elegidas</li> <li>Se ejecuta el juego y se controla según las direcciones del movimiento de la pieza mientras se ejecuta.</li> <li>Si se cumple la condición de derrota, se termina el juego, se escriben los datos de la partida en el archivo JUEGOS.TXT</li> </ol> </li> <li>Ver datos de partidas pasadas o de otros usuario <ol style="list-style-type: none"> <li>Pedir las opciones del usuario dentro del menú de los datos de juegos pasados. <ol style="list-style-type: none"> <li>Generar un diagrama de Gantt de los puntos de los jugadores de un país o de todos los jugadores</li> <li>Ver los reportes de datos de un jugador en específico al pedir sus datos correspondientes</li> <li>ver un reporte del top 5 jugadores de todos los países o de un país en particular.</li> </ol> </li> </ol> </li> <li>Salir del juego <ol style="list-style-type: none"> <li>Se asegura que el usuario realmente quiere salir del juego, y si lo está, cerrar el programa.</li> </ol> </li> </ol> </li></ol>	<ol style="list-style-type: none"> <li>Un mensaje en pantalla dependiendo del resultado del registro de usuario <ol style="list-style-type: none"> <li>Si los datos ingresados no son válidos (Proceso 2.1.2), se manda un mensaje de error dependiendo de su naturaleza</li> </ol> </li> <li>Un mensaje en pantalla dependiendo del resultado del inicio de sesión del usuario <ol style="list-style-type: none"> <li>Si los datos ingresados no existen y no se corresponden entre sí (Proceso 2.2.2) se manda un mensaje de error.</li> </ol> </li> <li>La pantalla del juego una vez se inicia y se concretan las configuraciones (Proceso 2.4.1) <ol style="list-style-type: none"> <li>El tablero de juego dibujado en pantalla con sus cambios respectivos dependiendo de las acciones del usuario (Proceso 2.4.3)</li> <li>La condición de derrota secundaria según el modo elegido en las configuraciones <ol style="list-style-type: none"> <li>El tiempo restante en forma de reloj si es por tiempo.</li> <li>La cantidad de movimientos restantes si es por movimientos</li> </ol> </li> <li>El dato de juego contrario a la condición de derrota secundaria según el modo elegido en las configuraciones <ol style="list-style-type: none"> <li>El tiempo que ha pasado desde el inicio de la partida en forma de reloj si es por movimientos</li> <li>La cantidad de movimientos utilizados si es por tiempo.</li> </ol> </li> <li>Un mensaje mostrando la denominación de los ODS a forma numérica <ol style="list-style-type: none"> <li>También una publicidad en forma de imagen de cada ODS.</li> </ol> </li> <li>Los datos del usuario actual a modo de mensaje en pantalla. <ol style="list-style-type: none"> <li>El nombre de usuario</li> <li>El nombre completo</li> <li>El país de origen</li> <li>Los puntos acumulados en juegos anteriores</li> <li>El contador de puntos del juego actual</li> </ol> </li> <li>La información de juegos anteriores o de otros usuarios dependiendo de la selección del usuario ( entrada 6) <ol style="list-style-type: none"> <li>Un diagrama de Gantt de los puntos de los jugadores de un país o de todos los jugadores dependiendo de la selección del usuario (Entrada 8)</li> <li>Un reporte de datos de un jugador específico (Entrada 7)</li> <li>Ver un reporte del top 5 jugadores de un país (Entrada 8) o de todos los países dependiendo de la selección del usuario.</li> </ol> </li> </ol> </li></ol>

## 2.- Descomposición modular del problema

# TETRIS



### 3.- Diseño del programa: presentar los algoritmos de cada módulo y del programa principal.

#### NOTAS:

- Esta lógica está diseñada para funcionar en el modo consola de Lazarus, por lo que puede estar sujeta a cambios futuros.
- Lo mostrado, según indicaciones del docente, son los análisis de cada módulo presentado con los procesos detallados al máximo posible, además del algoritmo en pseudocódigo de aquellos módulos que estén listos al momento de la entrega.

#### Análisis de cada Módulo:

##### Creación de los archivos de datos

Entrada	Procesos	Salidas
Ninguna	<p>Se verifica si existe el archivo que contiene los datos de los usuarios (JUGADORES.TXT) y el archivo que contiene los datos de partidas pasadas. (JUEGOS.TXT).</p> <p>Si no existe alguno, se crea el archivo correspondiente en la carpeta del programa.</p> <p>Si existía anteriormente, no sucede nada.</p>	Ninguna

##### Registro

Entrada	Procesos	Salidas
El Nombre de usuario del jugador.	Se piden los datos de entrada al usuario.	Un mensaje de suceso o de error en pantalla dependiendo del resultado de registro.
El Nombre completo del jugador	Se va al módulo de [verificación de datos del archivo	
La Clave de seguridad del jugador	“Jugadores.txt”] para revisar si el nombre de usuario ya existe.	
El Correo electrónico del jugador	Se verifica si el correo ingresado cumple con las condiciones	
El país de procedencia del	necesarias (Si contiene el	

jugador.	<p>carácter “Arroba” (@))</p> <p>Se verifica que la contraseña ingresada cumple con las condiciones necesarias.</p> <ul style="list-style-type: none"> <li>• Si posee entre 8 y 10 caracteres</li> <li>• Si es una combinación de números y letras en mayúscula y minúscula.</li> <li>• Si no contiene acentos o la letra Ñ o ñ.</li> <li>• Si contiene alguno de los siguientes caracteres especiales: igual (=), asterisco (*), guión (-), Guión bajo (_) o punto (.).</li> <li>• Si no contiene cualquiera de los demás caracteres especiales no mencionados en la condición anterior</li> <li>• Si no contiene más de 3 caracteres alfabéticos, numéricos ni especiales iguales de forma consecutiva.</li> </ul> <p>Si se cumplen los requisitos necesarios, se escriben los datos del usuario en el archivo JUGADORES.TXT y se muestra en pantalla un mensaje de suceso.</p> <p>Si no se cumplen los requisitos, se muestra en pantalla un mensaje de error.</p>	
----------	---	--

#### Inicio de sesión

Entrada	Procesos	Salidas
El Nombre de usuario del jugador.	Se piden los datos de entrada al usuario.	Un mensaje de suceso o de error en pantalla dependiendo del resultado del inicio de sesión.
La Clave de seguridad del jugador.	Se va al módulo de [verificación de datos del archivo	

	<p>“Jugadores.txt”] para revisar si los datos ingresados existen y se corresponden entre sí.</p> <p>Si los datos existen y se corresponden, Se mantiene la sesión dentro del programa y se muestra en pantalla un mensaje de suceso</p> <p>Si los datos no existen o no se corresponden, se muestra en pantalla un mensaje de error.</p>	
--	--	--

#### Verificación de datos en “Jugadores.txt”

Entrada	Procesos	Salidas
<p>El Nombre de usuario del jugador.</p> <p>La Clave de seguridad del jugador.</p>	<p>Dependiendo de dónde se utilice este módulo, se hace una acción u otra.</p> <ul style="list-style-type: none"> <li>• Si se viene del registro de usuario, se lee cada registro dentro del archivo “Jugadores.txt” y se verifica si el “Nombre de usuario” ingresado por el usuario ya existe. De existir, se envía un mensaje de error en pantalla.</li> <li>• Si se viene del inicio de sesión, se lee el registro correspondiente al “Nombre de usuario” ingresado y se revisa si se corresponde con la “Clave de seguridad” ingresada. De no existir el usuario o de no corresponderse los datos, se muestra un mensaje de error en pantalla.</li> </ul>	<p>Un mensaje de error dependiendo de su naturaleza.</p>

#### Configuración de la partida

Entrada	Procesos	Salidas
<p>La selección del modo de juego que se desea tener en la partida. (Contra reloj o por movimientos)</p> <p>El tiempo máximo de juego si es contra reloj.</p> <p>La cantidad de movimientos permitidos si es por movimientos.</p> <p>La selección de piezas a utilizar dentro del juego.</p>	<p>Se pide el modo de juego al usuario</p> <ul style="list-style-type: none"> <li>Si el modo de juego es por contra reloj <ul style="list-style-type: none"> <li>Se pide al usuario el tiempo máximo de la partida.</li> <li>Se guarda en una variable el tiempo ingresado (Tiempo_Juego)</li> </ul> </li> <li>Si el modo de juego es por movimientos <ul style="list-style-type: none"> <li>Se pide al usuario la cantidad de movimientos permitidos en la partida.</li> <li>Se guarda en una variable la cantidad de movimientos ingresados (Movimientos_Juego)</li> </ul> </li> </ul> <p>Se pide al usuario las piezas que se desean utilizar las partidas, hasta un máximo de 5 piezas.</p> <ul style="list-style-type: none"> <li>Se guarda en un arreglo los números que identifican las piezas elegidas (Piezas_Partida)</li> </ul>	<p>Ninguna.</p>

#### Desarrollo del juego o “bucle de juego”

Entrada	Procesos	Salidas
<p>La dirección de movimiento de la pieza actual (Derecha, izquierda o abajo)</p>	<ul style="list-style-type: none"> <li>Antes de comenzar el juego, se genera un código de juego aleatorio y único, el cual se almacena en el archivo “JUEGOS.TXT” junto al identificador/Usuario del jugador y la fecha y hora de juego.</li> <li>Se tiene un arreglo 3D que corresponde al tablero de juego, de 12x9x2 y de coordenadas [i,j,k] para facilidad en la explicación del proceso. <ul style="list-style-type: none"> <li>El arreglo [i,j,1] será llamado “Tablero real”, en dónde se irán a encontrar las piezas que ya hayan “caído” en</li> </ul> </li> </ul>	<p>El tablero de juego dibujado en pantalla, variando según los movimientos y las acciones del jugador.</p>

	<p>el tablero.</p> <ul style="list-style-type: none"> <li>○ El arreglo [i,j,2] será llamado “Tablero falso”, en donde únicamente se encontrará la pieza que esté “Cayendo” en ese momento.</li> <li>• Se tiene un arreglo constante 3D de 8x4x4 llamado “PiezasT”, que contiene los valores y la “Forma” respectiva de las 8 piezas elegibles. El valor 0 corresponde a “Ninguna pieza”</li> <li>• Se genera aleatoriamente una de las piezas seleccionadas por el usuario (Piezas_Partida) al asignar los valores contenidos en la posición respectiva del arreglo “PiezasT” en el “tablero falso”, específicamente iniciando el dibujado de la figura en la columna 4 (Aproximadamente la mitad del tablero).</li> <li>• Se asignan de 4 a 6 variables para los “Bordes” de la pieza dependiendo de su naturaleza (Borde superior, borde inferior, borde derecho, borde izquierdo, y dos bordes dinámicos que dependen de las “Hendiduras” que tienen algunas piezas. En específico, las que corresponden a los números 3, 4, 5, 7 y 8). <ul style="list-style-type: none"> <li>○ El borde superior siempre iniciará en 1 al ser la primera fila en la que se dibuja la pieza.</li> <li>○ El borde inferior corresponde a la fila del tablero en la que se encuentra el último tramo de la pieza dependiendo de su altura.</li> <li>○ El borde izquierdo siempre iniciará en 4 para todas las piezas, al ser la columna en la que se comienzan a dibujar todas las piezas.</li> <li>○ El borde derecho corresponde a la posición de la columna del tablero en la que se encuentra el último trazo de la pieza dependiendo de su longitud.</li> <li>○ Los bordes “Dinámicos” dependen de la posición de la columna del tablero en donde se encuentran las hendiduras correspondientes.</li> </ul> </li> <li>• La pieza va bajando en el tablero aproximadamente cada segundo al recorrer las filas dentro del “Tablero falso” dentro de los límites del “borde superior” y del “borde inferior”, y asignarle el valor que contenga en la fila superior a ella, y se suma en 1 al “borde superior e inferior” de la pieza para representar su bajada.</li> <li>• Se desplaza la pieza hacia los lados dependiendo de la entrada del movimiento elegido por el usuario (flecha Derecha/tecla D o flecha izquierda/tecla A) <ul style="list-style-type: none"> <li>○ Si es a la derecha, se recorren las columnas del “Tablero falso” dentro de los límites del “borde derecho” y el “borde izquierdo” de izquierda a derecha y se asignan los valores de cada columna a la que se encuentra a la izquierda de ella. También se suma en “1” a los bordes izquierdo y derecho.</li> <li>○ Si es a la izquierda, se recorren las columnas del</li> </ul> </li> </ul>	<p>El nombre de usuario, el nombre real y el país de origen del jugador</p> <p>Los puntos acumulados por el usuario de juegos anteriores</p> <p>El contador actualizado de puntos acumulados en el juego actual.</p> <p>El contador actualizado de movimientos usados o restantes en la partida.</p> <p>El reloj actualizado del tiempo pasado o restante en la partida.</p>
--	--	--



	<p>“Tablero falso” dentro de los límites del “borde derecho” y el “borde izquierdo” de izquierda a derecha y se asignan los valores de cada columna a la que se encuentra a la derecha de ella. También se resta en “1” a los bordes izquierdo y derecho.</p> <ul style="list-style-type: none"> <li>○ Al no haber “Fila a la derecha” o “Fila a la izquierda” para las filas 12 y 1 respectivamente (Al ser los bordes del tablero), se asignan a 0 directamente una vez ya no hay una pieza en ellas.</li> <li>○ Cada vez que se realiza un movimiento, se realiza una suma o resta dependiendo de la elección del usuario en el módulo [configuración de la partida] <ul style="list-style-type: none"> <li>• Si se eligió el modo con límite de movimientos, se resta en 1 a la cantidad de movimientos ingresada por el usuario.</li> <li>• Si se eligió el modo con límite de tiempo, se suma en 1 a la cantidad de movimientos, iniciando por 0 al comenzar la partida.</li> <li>• Posteriormente se muestra el contador de movimientos actualizado en pantalla.</li> </ul> </li> <li>• Cuando se detecta una colisión horizontal con algún borde del tablero o alguna pieza, no se permitirá ningún movimiento a la dirección de la colisión. Esto al revisar si en las columnas del “Tablero real” a la derecha o izquierda de los bordes derechos o izquierdos (Y los bordes dinámicos si los tiene) se encuentra alguna pieza, o bien, si la pieza se encuentra en la columna 1 o 12 (Bordes de tablero)</li> <li>• Se desplaza hacia abajo la pieza dependiendo de la entrada del movimiento elegido por el usuario (flecha abajo/Tecla S), esto al verificar la última fila libre en el tablero dependiendo de la forma de la pieza y las que ya se encuentren colocadas, y se envía automáticamente la actual hacia esa fila para posicionarla. Esto es para hacer más dinámico o veloz el desarrollo de las partidas.</li> <li>• Cuando se detecta una colisión con el borde inferior de la cuadrícula, o con otra pieza en el “Tablero real”, se “Copiar” los datos del “Tablero falso” en el “Tablero real” (únicamente los datos que no son “Cero”), y se reinicia el bucle de juego, limpiando en ceros el “tablero falso” y asignando una nueva pieza.</li> </ul>	
--	---	--

	<ul style="list-style-type: none"> <li>• Cada segundo mientras el juego esté corriendo, se realiza una suma o resta dependiendo de la elección del usuario en el módulo [configuración de la partida] <ul style="list-style-type: none"> <li>○ Si se eligió el modo con límite de movimientos, se suma en 1 a la cantidad de segundos pasados, iniciando en 0 al comenzar la partida.</li> <li>○ Si se eligió el modo con límite de tiempo, se resta en 1 a la cantidad de movimientos ingresada al usuario.</li> <li>○ Posteriormente se muestra el contador de segundos en pantalla.</li> </ul> </li> <li>• Cuando una fila del “Tablero real” esté llena, se eliminará, se moverán todos los elementos de la cuadrícula por encima de la misma hacia abajo, y se calculará la puntuación, en base al “valor en puntos” de los trozos de las piezas que estaban en la fila a eliminar. <ul style="list-style-type: none"> <li>○ El puntaje se va acumulando en una variable y se muestra en pantalla.</li> </ul> </li> <li>• Dependiendo de cuál sea la pieza en movimiento, un mensaje e imagen relacionados a su “objetivo-ODS” aparecerá en pantalla.</li> <li>• El juego termina una vez se cumple la condición de derrota principal o la condición de derrota secundaria seleccionada en el módulo de [configuración de la partida] <ul style="list-style-type: none"> <li>○ La condición de derrota principal es que en el “tablero real” se envíe una pieza desde el “Tablero falso” que al momento de caer se encuentre en el tope de la matriz</li> <li>○ La condición de derrota secundaria es aquella elegida por el usuario, es decir, que se acabe el tiempo de partida, o se acaben los movimientos permitidos en la partida. (Sean 0 en total)</li> </ul> </li> </ul>	
--	--	--

#### Fin del juego

Entrada	Procesos	Salidas
Ninguna	Se muestra una pantalla de derrota junto al puntaje obtenido por el jugador.	El mensaje de “JUEGO CONCLUIDO”  La cantidad de puntos obtenidos en el juego.

#### Escritura de datos en “Juegos.txt”

Entrada	Procesos	Salidas
---------	----------	---------

Ninguna	<p>Cuando se concluye un juego, se escriben en el archivo de texto "JUEGOS.TXT" los datos de la partida que recién acaba de terminar, esto en la posición ya creada al inicio del juego, junto al código único de la partida.</p> <ul style="list-style-type: none"> <li>• El código del juego</li> <li>• El identificador/Usuario del jugador</li> <li>• La cantidad de puntos ganados</li> </ul>	Ninguna
---------	--	---------

#### Verificación del registro de puntajes

Entrada	Procesos	Salidas
La selección de aquello que se quiere verificar.	<p>Dependiendo de la selección, se ejecutará el módulo correspondiente</p> <ul style="list-style-type: none"> <li>• [Vista del historial de partidas de un jugador]</li> <li>• [Vista de un diagrama de Gantt del puntaje de los jugadores de un país o de todos los países.]</li> <li>• [Vista del top 5 jugadores de un país o de todos los países.]</li> </ul>	Ninguna

#### Vista del historial de partidas de un jugador

Entrada	Procesos	Salidas
El usuario del jugador del que se quiere ver su historial.	<p>Se pide el dato de entrada al usuario</p> <p>Se verifica si el nombre del usuario se encuentra dentro de la lista de jugadores en el archivo "JUGADORES.TXT". Sino, se muestra en pantalla un mensaje de error en el que se menciona que no existe el usuario.</p> <p>Si el usuario existe, se verifica si existen partidas realizadas por el</p>	<p>Los datos básicos del jugador elegido (Nombre de usuario, país de origen, nombre completo y puntuación total sumando el de todas las partidas realizadas.)</p> <p>El historial de partidas del jugador elegido, presentando el código de la partida, la fecha, hora y los puntos obtenidos en cada una.</p> <p>Un mensaje de error</p>

	<p>mismo en el archivo "JUEGOS.TXT". Sino, se muestra en pantalla un mensaje de error en el que se menciona que el usuario no tiene partidas.</p> <p>Si el usuario tiene partidas grabadas, se muestran sus datos básicos y su historial de partidas en pantalla.</p>	dependiendo de su naturaleza.
--	---	-------------------------------

Vista de un diagrama de Gantt del puntaje de los jugadores de un país o de todos los países.

Entrada	Procesos	Salidas
<p>La selección del usuario del rango del diagrama de Gantt (de un país o de todos los países)</p> <p>El país del que se desea ver el diagrama de Gantt de ser ese el caso.</p>	<p>Se piden los datos de entrada correspondientes al usuario</p> <p>Se va al módulo de [lectura de datos en "juegos.txt"]</p> <p>Se revisan los puntajes totales de cada usuario dentro del rango elegido y se genera el diagrama de Gantt.</p> <p>Se muestran los datos de salida.</p>	<p>El diagrama de Gantt de las puntuaciones de los usuarios dentro del rango elegido</p>

Vista del top 5 jugadores de un país o de todos los países.

Entrada	Procesos	Salidas
<p>La selección del usuario del rango del top 5 (De un país o de todos los países)</p> <p>El país del que se desea ver el Top 5 de ser ese el caso.</p>	<p>Se piden los datos de entrada correspondientes al usuario</p> <p>Se va al módulo de [lectura de datos en "juegos.txt"]</p> <p>Se clasifican los datos de los puntajes totales de todos los usuarios y se clasifican para generar el top 5.</p> <p>Se muestran los datos de salida.</p>	<p>El top 5 de los jugadores dentro del rango elegido.</p>

Lectura de datos en «Juegos.txt»

Entrada	Procesos	Salidas
Ninguna	Independientemente de donde se venga hacia este módulo, se leen todos los datos de las partidas guardadas en "Juegos.txt".	Ninguna.

**Algoritmo en pseudocódigo de los módulos ya completados.**

## 1) Módulo de Registro de Jugadores:

**Procedimiento** RegistroJugadores;

### Declaraciones

Lista\_Paises\_LATAM: Arreglo [1..20] de tipo Cadena [20] = ("Argentina", "Bolivia", "Brasil", "Chile", "Colombia", "Costa Rica", "Cuba", "Ecuador", "El Salvador", "Guatemala", "Haiti", "Honduras", "Mexico", "Nicaragua", "Panama", "Paraguay", "Peru", "Republica Dominicana", "Uruguay", "Venezuela");

Encontrado, Clave\_Min, Clave\_enie, Clave\_Esp, Clave\_May, Cad\_Repe, arroba: Boleano;

NombreR, UsuarioR, CorreoR, Busqueda, Palabra: Cadena [40];

ClaveR, Aux\_Cad: Cadena [10];

Cont\_CadRepe, Opcion\_LATAM, i, k, Inicio\_Rep\_Paises, Fin\_Rep\_Paises: Entero;

### Cuerpo del procedimiento

#### Inicio

**//\*\*\*\*\*Iniciando las variables\*\*\*\*\*//**

i <- 0;

k <- 0;

Inicio\_Rep\_Paises <- 1;

Fin\_Rep\_Paises <- 4;

**//\*\*\*\*\*Verificando que el nombre del usuario se encuentre disponible\*\*\*\*\*//**

Repetir

Encontrado <- Falso;  
Color de texto (Blanco);  
Escribir ("Usuario: ");  
Leer (UsuarioR);  
Abrir Archivo en modo lectura (jugadores);

Repetir mientras (No se halla leído todo el archivo (jugadores)) Hacer

Leer (Jugadores, busqueda);  
Si (posicion (UsuarioR, Busqueda) <> 0) Entonces  
Encontrado<- Verdadero;  
Fin-Si;  
Fin-RM;  
Si (Encontrado= Verdadero) Entonces  
Color de texto (rojo);  
Escribir ("Usuario no disponible. Intente otro.");  
Fin-Si;  
Cerrar (jugadores);

Hasta (Encontrado = Falso);

**//\*\*\*\*\*Registrando el nombre y apellido de los jugadores\*\*\*\*\*//**

Color de texto (Blanco);  
Escribir ("Nombre y Apellido: ");  
Leer (NombreR);

**//\*\*\*\*\*Verificando que el correo cumpla con los requisitos\*\*\*\*\*//**

Repetir

Escribir ("Correo Electronico: ");  
Leer (CorreoR);

Repetir para i, 1, longitud (CorreoR), +1, Hacer

Si (CorreoR[i] = "@") Entonces

Arroba <- Verdadero;

Fin-Si;

Fin-RP;

Si (Arroba = falso) Entonces

Escribir ("Ingrese una dirección de correo valida.");

Fin-Si;

Hasta (Arroba= Verdadero);

**//\*\*\*\*Verificando que la contraseña cumpla con todos los requisitos\*\*\*\*//**

Cont\_CadRepe <- 1;

Clave\_enie <- Falso;

Clave\_Esp <- Falso;

Clave\_May <- Falso;

Clave\_Min <- Falso;

Cad\_Repe <- Falso;

arroba <- Falso;

Repetir

Color de texto (Blanco);

Escribir ("Requisitos para su clave:");

Escribir ("1. Debe poseer al menos una (1) letra mayuscula.");

Escribir ("2. Debe poseer al menos una (1) letra minuscula.");

Escribir ("3. Debe poseer al menos uno de los siguientes caracteres especiales (=, \*, -, \_,.).");

Escribir ("4. No debe poseer una enie.");

Escribir ("6. No debe tener más de 4 caracteres iguales (aaaa, 1111).");

Escribir ("Clave: ");

Leer (ClaveR);

**//\*\*\*\*Verificando si la contraseña posee más de 10 caracteres\*\*\*\*\*//**

Si (longitud (ClaveR) < 10) Then

**//\*\*\*\*Recorriendo cada carácter de la contraseña\*\*\*\*\*//**

Repetir para k,1, longitud (ClaveR), +1, Hacer

**//\*\*\*\*\* Verificando si la contraseña posee letras minúsculas\*\*\*\*\*//**

Si (ClaveR[k] Esta ["a".."z"]) y (Clave\_Min=falso) entonces

Clave\_Min <- Verdadero;

Fin-Si;

**//\*\*\*\*\* Verificando si la contraseña posee letras mayúsculas\*\*\*\*\*//**

Si (ClaveR[k] Esta ["A".."Z"]) Y (Clave\_May= falso) Entonces

Clave\_May <- Verdadero;

Fin-Si;

**//\*\*\*\*\* Verificando si la contraseña posee caracteres especiales\*\*\*\*\*//**

Si (ClaveR[k] Esta ["=", "\*", "-", "\_", "."]) Y (Clave\_Esp=falso) Entonces

Clave\_Esp <- Verdadero;

Fin-Si;

**//\*\*\*\*\* Verificando si la contraseña posee enies\*\*\*\*\*//**

Si (ClaveR [k] Esta [#164, #165]) Y (Clave\_enie=falso) Entonces

Clave\_enie <- Verdadero;

Fin-Si;

**//\*\*\*\*\*Verificando si se repite un carácter más de 4 veces\*\*\*\*\*//**

Si (ClaveR[k] esta ['a'..'z', 'A'..'Z', '=', '\*', '-', '\_', '.', '0'..'9']) Entonces

Si (ClaveR[k]=Aux\_Cad) Entonces

Cont\_CadRepe <- Cont\_CadRepe + 1;

Fin-Si;



```
Si (ClaveR[k]<>Aux_Cad) Entonces
  Cont_CadRepe <- 1;
  Aux_Cad <- ClaveR[k];
Fin-Si;
Fin-Si;
```

```
Si (Cont_CadRepe >= 4) entonces
  Cad_Repe <- True;
Fin-Si;
```

```
Fin-RP
Fin-Si;
```

```
Si (Clave_Min=falso) O (Clave_May=falso) O (Clave_Esp=falso) O (Clave_enie=Verdadero)
O (Cad_Repe = Verdadero) Entonces
```

```
Color de texto (Rojo);
Escribir ("Su Clave no cumple con todos los requisitos. Intente nuevamente.");
Fin-Si;
```

```
Hasta (Clave_Min=Verdadero) y (Clave_May=Verdadero) y (Clave_Esp=Verdadero) y
(Clave_enie=Falso) y (Cad_Repe =Falso);
```

**//\*\*\*\*\*Selección del pais nativo del jugador\*\*\*\*\*//**

```
Escribir ("Indique su país: ");
Repetir mientras (Fin_Rep_Paises <= 20) Hacer
  Repetir Para I, Inicio_Rep_Paises, Fin_Rep_Paises, +1, Hacer
    Escribir (i, ". ", Lista_Paises_LATAM[i], " ");
  Fin-RP;
  Escribir ();
  Inicio_Rep_Paises <- Fin_Rep_Paises;
  Fin_Rep_Paises <- Fin_Rep_Paises + 4;
Fin-RM;
Leer (Opcion_LATAM);
```

**//\*\*\*\*\*Guardando la información en el archivo\*\*\*\*\*//**

Adjuntar al final del archivo (jugadores);  
Escribir (Jugadores, "JUGADOR: ");  
Escribir (Jugadores, UsuarioR, " ");  
Escribir (Jugadores, ClaveR, " ");  
Escribir (Jugadores, NombreR, " ");  
Escribir (Jugadores, CorreoR, " ");  
Escribir (Jugadores, Lista\_Paises\_LATAM[Opcion\_LATAM], " ");  
Cerrar (jugadores);  
Color de texto (Verde);  
Escribir ("Registro Completado con éxito.");

**Fin del procedimiento** RegistrodeJugadores;

## **2) Módulo de inicio de sesión de los jugadores:**

**Procedimiento** IniciodeSesion;

### **Declaraciones**

Encontrado\_clave, Encontrado\_usuario: Boleano;  
IS\_Clave, IS\_Usuario, busqueda, Palabra: Cadena [40];  
i: Entero;

### **Cuerpo del procedimiento**

#### **Inicio**

i <- 0;  
Encontrado\_clave <- Falso;  
Encontrado\_usuario <- Falso;  
Repetir  
Color de texto (Blanco);  
Abrir en modo de lectura (Jugadores);  
Escribir ("Usuario: ");  
Leer (IS\_Usuario);

Escribir ("Clave: ");

Leer (IS\_Clave);

Repetir mientras (no se haya leído todo el archivo (Jugadores)) hacer

Leer (jugadores, busqueda);

Si (posicion (IS\_Usuario, Busqueda) <> 0) Entonces

Encontrado\_Usuario <- Verdadero;

Fin-Si;

Si (posicion (IS\_Clave, Busqueda) <> 0) Entonces

Encontrad\_Clave<- Verdadero;

Fin-Si;

Fin-RM

Si (Encontrado\_usuario= Verdadero) y (Encontrado\_clave= Verdadero) entonces

Color de texto (Verde);

Escribir ("inicio de sesion exitoso");

De lo contrario

Color de texto (Rojo);

Escribir ("Usuario o Clave Incorrecto. Vuelva a intentarlo.");

Fin-Si

Cerrar (jugadores);

Hasta (Encontrado\_usuario = Verdadero) y (Encontrado\_clave= Verdadero);

**Fin del Procedimiento** Iniciodesesion;

### 3) **Módulo de Creación de los archivos de datos**

**Procedimiento** Archivo;

**Declaraciones del procedimiento:**

Jugadores: texto;

## **Cuerpo del procedimiento**

### **Inicio**

Asignar (Jugadores, "Jugadores.TXT");  
Si no (existe el fichero ("Jugadores.TXT")) Entonces  
    Rehacer (Jugadores);  
Fin-si;  
Asignar (Juegos, "Juegos.TXT");  
Si no (existe el fichero ("Juegos.TXT")) Entonces  
    Rehacer (Juegos);  
Fin-si;

**Fin del procedimiento** Archivo;

## **4) Cuerpo del programa principal**

### **Inicio**

**//\*\*\*\*\*Inicialización del fichero\*\*\*\*\*//**

Llamar Archivo;

**//\*\*\*\*\*Pantalla de inicio\*\*\*\*\*//**

Repetir

    Color de texto (Blanco);  
    Escribir ("PROYECTO T");  
    Escribir ("1. Iniciar Sesión");  
    Escribir ("2. Registrarse");  
    Escribir ("3. Extras");  
    Escribir ("4. Salir del juego");  
    Leer (Opcion);

En caso de opción Hacer

**/\*\*Si se selecciona la opción 1, se abrirá un menu de inicio de sesión\*\*//**

1:

Llamar IniciodeSesion;

Fin-1;

**/\*\*Si se selecciona la opcion 2, se abrira un menu de Registro\*\*//**

2:

Llamar RegistrodeJugadores;

Fin-2;

Fin-Caso;

Hasta (Opcion = 4);

**Fin.**